

REMARKS

This Amendment is submitted in response to the Examiner's Action mailed August 17, 2005, with a shortened statutory period of three months set to expire November 17, 2005. Claims 1, 2, 5, 6, 8-10, 13, 14, 16-18, 21, 22, and 24 are currently pending.

The Examiner rejected claims 1-2, 5, 8-10, 13, 16-18, 21, and 24 under 35 U.S.C. § 102(b) as being anticipated by "Web Server Support for Tiered Services", published by *Bhatti*, 1999. This rejection is respectfully traversed.

Claims 1, 9, and 17 describe completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues. The Examiner states that *Bhatti* teaches these features in Figure 7, page 67 regarding the connection manager, and page 68 regarding the requesting scheduling.

Bhatti does not teach completing processing of multiple HTTP requests before beginning processing of other multiple requests. *Bhatti* teaches that after a single request has been selected, it will run until completion. This is a single request. *Bhatti* teaches that once a single request has been selected, it will complete. *Bhatti* does not teach that multiple requests will complete processing before beginning the processing of other multiple requests.

The Examiner refers to *Bhatti* as teaching these features at Figure 7, page 67, and page 68, lines 1-14. Figure 7 teaches multiple tiers. Requests are classified into these tiers.

Page 68, lines 1-14 teaches request scheduling. After requests are classified, they are scheduled for processing. This section of *Bhatti* teaches several different methods for how requests might be scheduled for processing. For example, requests might be scheduled such that that one class gets more requests scheduled than another class. This section of *Bhatti* does not anticipate Applicants' claims, however, because request scheduling is not what is claimed by Applicants.

Applicants claim completing the processing of multiples ones of HTTP requests that are stored in a first queue before beginning the processing of multiple ones of HTTP requests that are stored in a second queue. Scheduling a request is not the same as processing a request. A request could be schedule but then never processed.

One or more requests might be scheduled for processing but then be unable to complete processing, even though they were scheduled for processing, for some reason. For example, several requests could be scheduled for processing. Before starting the processing of these scheduled requests, the computer could crash. These scheduled requests would then not complete processing. Therefore, it is clear that scheduling and processing requests are different. Applicants claim processing requests, not scheduling requests. The Examiner is referring to a section of the reference that describes scheduling requests.

Further, Applicants claim completing processing of multiple requests. When *Bhatt* does discuss processing a request, *Bhatti* clearly describes processing a single request, not completing processing of multiple requests. *Bhatt* does not teach completing processing of multiple requests.

Further, *Bhatti* itself makes it clear that scheduling and processing are not the same. *Bhatti* teaches that a request is selected for processing after it has been scheduled. Scheduling a request and processing a request are not the same. See page 68, line 6-7.

Bhatti also clearly teaches that after a request has been selected for processing, it will run until completion. Page 68, lines 11-13. This section of *Bhatti* teaches that one request will complete processing once it has started processing. This section of *Bhatti* does not teach that multiple requests will complete processing before beginning processing of other multiple requests. Applicants claim completing processing of multiple ones of HTTP requests in a first queue before beginning processing of multiple ones of HTTP requests in a second queue. By teaching that one request will run until completion, *Bhatti* does not teach completing processing of multiple requests before beginning processing of other multiple requests.

Applicants' claims also describe specifically where these multiple requests are located. Applicants' claims describe completing processing of multiple ones of HTTP requests that are stored in a first queue before beginning processing of multiple ones of

HTTP requests that are stored a second queue. *Bhatti* does not teach completing processing of multiple requests that are stored in a first queue before beginning processing of multiple requests that are stored in a second queue.

Applicants' claims describe completing the processing of the multiple requests in the first queue before beginning the processing of multiple requests in the second queue regardless of whether or not the first one of the plurality of priorities is a higher priority than the second one of the plurality of priorities. Nothing in *Bhatti* teaches this feature. The Examiner has not referred to any section of *Bhatti* that teaches regardless of whether or not the first one of the plurality of priorities is a higher priority than the second one of the plurality of priorities.

Because *Bhatti* does not describe, teach, or suggest (1) completing processing of multiple requests before beginning the processing of other multiple requests, (2) the feature of completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues, or (3) the feature of regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities, *Bhatti* does not anticipate Applicants' claims.

The remaining claims that depend from the independent claims discussed above are also not anticipated by *Bhatti*. Because *Bhatti* does not teach completing processing of multiple requests before beginning the processing of other multiple requests, completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues, or completing the processing of multiple requests in the first queue before beginning the processing of multiple requests in the second queue regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities, *Bhatti* does not anticipate the features of these dependent claims.

The Examiner rejected claims 6, 14, and 22 under 35 U.S.C. § 103(a) as being unpatentable over *Bhatti* in view of U.S. Patent 6,769,019 issued to *Ferguson*. This rejection is respectfully traversed.

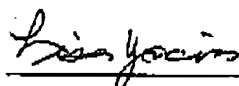
The Examiner stated that *Bhatti* does not teach determining whether there is a backlog of pending HTTP requests waiting to be processed by said application, and in response to a determination that there is no backlog, immediately processing said one of said plurality of HTTP requests. The Examiner relies on *Ferguson* to supply the features believed missing from *Bhatti*.

None of the cited references teaches completing processing of multiple HTTP requests before beginning the processing of other multiple HTTP requests, completing processing of multiple HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple HTTP requests that are stored in said second one of said plurality of different queues, or completing the processing of the requests in the first queue before beginning the processing of the requests in the second queue and regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities. Thus, the combination of *Bhatti* and *Ferguson* does not describe, teach, or suggest determining whether there is a backlog of pending HTTP requests waiting to be processed by said application, and in response to a determination that there is no backlog, immediately processing said one of said plurality of HTTP requests in combination with completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues and regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities. Therefore, the combination of cited references does not render Applicants' claims unpatentable.

Applicants' claims are believed to be in a patentable form. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 10.17.05

Respectfully submitted,



Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants